

TED Talks Comments Sentiment Classification Using Machine Learning Algorithms

Maria El-Badaoui^{*}, Noredine Gherabi, Fatima Quanouni

National School of Applied Sciences, Sultan Moulay Slimane University, Lasti Laboratory, Khouribga 25000, Morocco

Corresponding Author Email: maria.el-badaoui@usms.ac.ma



Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ria.380315>

ABSTRACT

Received: 1 December 2023

Revised: 3 March 2024

Accepted: 7 June 2024

Available online: 21 June 2024

Keywords:

sentiment analysis (SA), TED talks, Textblob, Machine Learning (ML), Random Forest (RF), SVM, KNN, Naïve Bayes (NB), Decision tree

This research presents a comparative analysis of sentiment analysis techniques applied to user comments on YouTube, with a specific focus on TED talks. The proliferation of social media platforms has provided individuals with unprecedented opportunities to express their opinions and emotions. YouTube, as a leading video-sharing platform, has become a significant hub for user-generated content and discussions on a wide range of topics. In light of the exponential growth of unstructured and semi-structured data, sentiment analysis plays a critical role in extracting valuable emotional insights from online interactions. To evaluate sentiments expressed in YouTube comments, a self-created and meticulously labeled dataset comprising user comments was employed. The study compared the performance of five ML techniques: NB, SVM, RF, KNN, and DT. The performance of the classifiers was evaluated using key evaluation metrics such as Precision, Recall, and F1-score. The findings of this research offer valuable insights into the efficacy of various machine learning techniques for sentiment analysis in the context of YouTube comments on TED talks. Among the classifiers, SVM demonstrated the highest Precision, Recall, and F1-score, indicating its effectiveness in accurately identifying sentiment in YouTube comments. Random Forest and Decision tree also displayed competitive performance, while KNN and Naïve Bayes exhibited slightly lower accuracy. These results provide researchers and practitioners with valuable information to make informed decisions regarding the selection of appropriate ML techniques for SA tasks on social media platforms.

1. INTRODUCTION

Sentiment analysis is a field of study that involves extracting and analyzing emotional information from online inputs such as social media comments, product reviews, and customer feedback. It has become an increasingly popular research area due to the growing availability of data in unstructured or semi-structured formats.

One area where sentiment analysis could prove valuable is in the context of TED talks. TED events bring together people from different walks of life to share ideas and stories that can inspire and motivate audiences. The comments and feedback left by viewers on TED talks can provide valuable insights into the overall sentiment of the audience towards the speaker and their message.

The aim of this research is to identify the sentiment polarity of comments made on TED Talks., i.e., whether they express positive, negative, or neutral emotions, using sentence-level sentiment analysis. The comments were classified using various ML techniques on the same dataset, and the accuracy of each approach was assessed. The classifiers employed in this study are Naïve Bayes, Support Vector Machine, Decision Trees, Random Forest, and K Nearest Neighbors (KNN).

The first step is to collect data of TED talks comments, one would typically need to scrape the TEDx website or use an API to access the comments section of each TED talks. The

comments then need to be extracted in a structured format, such as a CSV or JSON file, to allow for analysis. After collecting the data, it must be cleaned and preprocessed to remove duplicates and irrelevant information and prepare the text data for sentiment analysis. The preprocessing includes stop word removal, stemming tokenization and part-of-speech tagging. Finally, the preprocessed data can be used to train ML models for SA, and their performance can be evaluated using metrics like accuracy, precision, recall, and F1-score.

The structure of the paper is as follows: Related works are discussed in Section II, Section III outlines the proposed methodology, Section IV gives a concise overview of the classifiers employed, Section V provides the result analysis, and Section VI presents the conclusions.

2. RELATED WORK

Numerous research studies have been conducted in the domain of text classification and SA. The primary challenge in SA is to determine whether a given text is positive or negative or neutral. There are three categories of sentiment polarity classification: entity level, document level, and sentence level.

The goal of entity sentiment analysis is to analyze sentiments expressed towards a particular entity in a document, such as a person, party, or institution. The aim is to detect the emotion

words that are directly linked to and participate in the entity, to accurately reveal the feeling it inspires.

The document level evaluates the overall sentiment [1-3], that is, whether the entire text has negative or positive valence.

The sentence level deals with the sentiment classification of individual sentences.

Scholars from various parts of the world have employed semi-supervised, unsupervised, and supervised machine learning techniques in their research as mentioned in Figure 1.

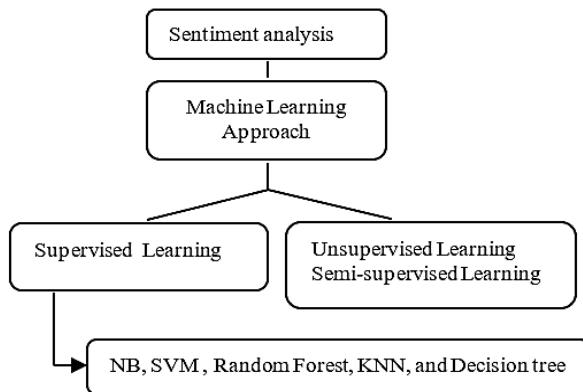


Figure 1. Approaches to sentiment analysis

Al Amrani et al. [4] proposed a hybrid approach for sentiment analysis in product reviews. They combined Random Forest and SVM algorithms to analyze the sentiments expressed in these reviews. The study utilized a dataset of product reviews obtained from Amazon and applied various text pre-processing techniques. This hybrid approach was then evaluated and compared to individual classifiers, taking into account important metrics such as accuracy, precision, recall, and F-measure. The findings of the study demonstrated the effectiveness of the hybrid approach in superior achievement.

Srujan et al. [5] used a variety of preprocessing approaches and classifiers to categorize book reviews as either positive or negative. The primary purpose of this study is to compare the accuracy and processing time of these classifiers. Furthermore, they focused their study on comparing the sentiment scores linked with various works. The sentiments under examination fall into two broad categories, positive and negative, and are accompanied by eight fundamental emotions.

The aim of this study [6] is to compare classifications of e-commerce customer opinions on Tokopedia and Bukalapak on Twitter using text mining techniques to analyze customer sentiments and compare the results of three different classification approaches (Decision Tree, K-Nearest Neighbor and Naïve Bayes Classifier) to find the best precision. The Naïve Bayes approach gives the highest results, with a precision of 88.50% and a recall of 64%.

Tripathy et al. [7] utilized supervised learning methods to classify reviews based on their polarity. The recommended approach involved four main steps. Firstly, preprocessing was performed to remove stop words, numeric values, and special characters. Next, the text reviews were transformed into a numerical matrix. Subsequently, these matrices were used as input for four distinct classifiers. The evaluation of the approach was conducted by classifying two datasets, and numerous metrics like precision, recall, F-measure, and classification accuracy were calculated. The results demonstrated that the Random Forest classifier outperformed

the other classifiers when applied to the polarity in IMDB datasets.

Amini Motlagh et al. [8] explored the sentiment analysis of tweets using various data mining classifiers. They applied K-NN, DT, SVM, and NB to analyze tweet sentiments on two datasets: a binomial dataset (positive and negative) and a polynomial dataset (positive, negative, and neutral). The findings showed that SVM consistently outperformed the other algorithms, with accuracy improvements of 3.53% and 7.41% for the two-class and three-class datasets, respectively.

This research [9] presented a study on sentiment analysis of Twitter users towards ChatGPT, an AI-powered chatbot. By collecting tweet data from 5000 users and applying the Naive Bayes algorithm. The analysis concluded that ChatGPT was highly regarded by the most of users, with an accuracy rate of 80%.

Gupta et al. [10] used a massive unlabeled review dataset collected from Amazon, IMDB, and Yelp. To examine the sentiment in the reviews, the authors used a variety of ML models such as: SVM, Random Forest, Multinomial Naive Bayes, and KNN and feature extraction approaches (TF-IDF and bag of words). The Random Forest classifier produces the best results in the investigation, with an accuracy of more than 78%.

Saha et al. [11] employed Textblob for data pre-processing, as well as for calculating polarity and confidence. To validate their findings, they conducted experiments using SVM and Naïve Bayes in Weka. The results indicated that Naïve Bayes achieved a higher accuracy rate of 65.2%, surpassing the accuracy rate of SVM by 5.1%.

The researchers [12] labeled the emails using VADER sentiment and a Swedish sentiment lexicon. Then, they used these labels to understand the emotions in the emails by training two computer models known as Support Vector Machines. In their experiments, one of the models, called LinearSVM, performed well. It achieved an average score of 0.834 for determining emotions in emails and an average score of 0.896 overall. Additionally, it could predict how people would feel in the next email in the conversation thread. Its average score for this prediction was 0.688, and its overall average score was 0.805.

Other approaches employ deep learning to classify sentiment polarity, as demonstrated by Ain et al. [13] who applied DNN, CNN, and Deep Belief Network (DBN), to address SA tasks such as sentiment classification, cross-lingual challenges, and product review analysis.

Habimana et al. [14] examined deep learning techniques on specific datasets and suggested that adding models like BERT, sentiment-specific word embedding models, cognitive-based attention models, and commonsense knowledge can improve performance.

This study [15] proposed a hybrid model called BERT-BiLSTM-CNN that integrates BERT, BiLSTM, and a triple parallel CNN branch and achieves exceptional accuracy when analyzing sentiments on Turkish tweets.

This research [16] introduces a deep learning approach aimed at analyzing sentiments expressed on Twitter concerning Higher education distance learning. They utilized Twint to gather 24,642 tweets. The obtained tweets passed through pre-processing and were classified as neutral, negative, and positive before being sent to the suggested Bi-LSTM model that utilizes self-attention and Glove word embedding to extract features. Using the Adam optimizer, this model outperformed the LSTM, Bi-LSTM, and CNN-Bi-LSTM

models, with the highest test accuracy of 95%.

3. METHODOLOGY

3.1 Twitter data VS. YouTube data

Twitter, a microblogging and social networking platform, enables users to share and express opinions through concise messages. In contrast, YouTube serves as a video-sharing website where users can upload videos and control access through privacy settings. Additionally, users can post comments and reviews on the videos they watch. Users on both platforms tend to employ informal language, disregarding conventional grammar and spelling rules. Their posts often feature emoticons, texting-style abbreviations, and repetitive letters or punctuation for emphasis.

The main difference is in the type of content that is created. On Twitter, users create brief pieces of information known as "tweets" with a maximum limit of 140 characters. These tweets cover a diverse range of topics, including opinions on personalities, politicians, products, companies, and events. Twitter users also employ certain symbols that transcend language barriers. For instance, the "@" symbol is used for mentioning other users, while the "#" (hash tag) is employed to mark topics or keywords, making messages more visible to others. The writing style on Twitter is characterized by brevity and speed, often incorporating acronyms and emoticons to convey opinions concisely.

On YouTube, users primarily offer reviews on the content of videos. Unlike Twitter's character limit, there are no constraints on the length of reviews or comments. These posts solely focus on analyzing and discussing the video content and lack specific symbols or conventions similar to those used on Twitter.

3.2 Proposed methodology

The first aim of our system is to identify the polarity of sentiment in TED talks comments. The proposed system's architecture is depicted in Figure 2.

Our process begins with the acquisition of a dataset consisting of comments. Subsequently, our data is annotated using the polarity score provided by TextBlob. The resulting labeled dataset undergoes preprocessing to eliminate any noise or redundant information. Feature extraction techniques are then applied to extract relevant features from the dataset, which are used to train ML models. The performance of these models is assessed using evaluation metrics.

a) Step 1: Collecting Data

The first step in our methodology involves collecting comments from websites and YouTube, a crucial task in building a dataset for sentiment analysis. Websites like TED talks frequently feature comment sections where users share their thoughts and opinions about the talks. Similarly, YouTube provides a platform for users to engage with video content through comments.

To collect comments from these platforms, we employed various methods. Initially, manual extraction by copying and pasting comments into a text file was considered. However, due to the time-consuming nature of this approach, we opted for web scraping techniques to streamline the data collection process.

Web scraping involves automatically extracting data from websites using bots or crawlers. This method offers efficiency and scalability, allowing us to collect a large number of comments quickly. We extracted specific elements from websites, including comments and reviews, using BeautifulSoup, a Python library that parses HTML and XML documents. By leveraging BeautifulSoup's capabilities, we were able to gather comments from TED talks videos on YouTube effectively.

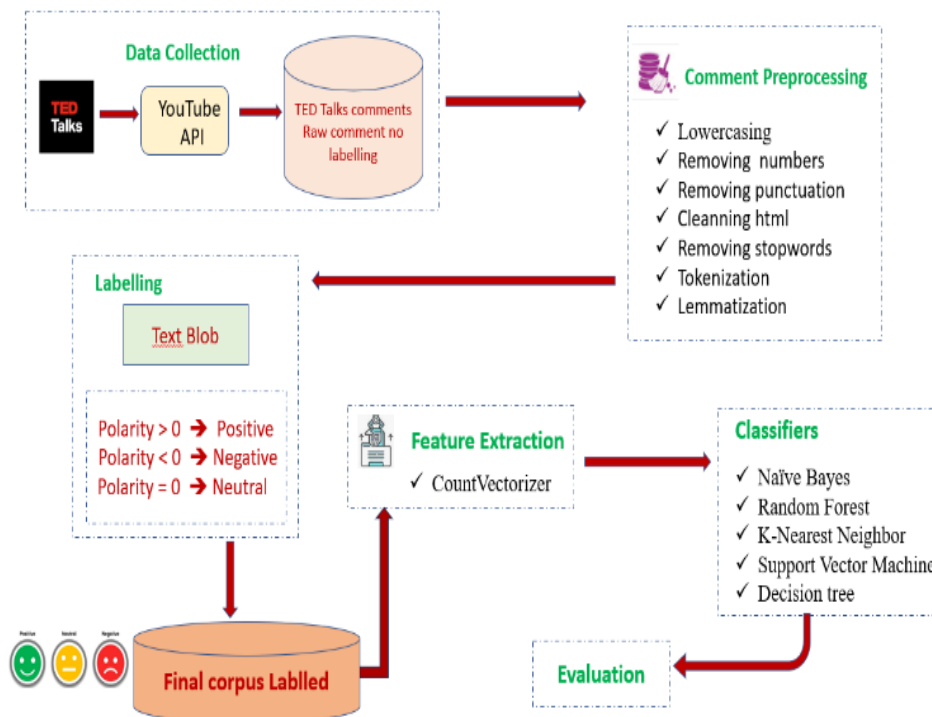


Figure 2. Workflow of the proposed methodology for TED talks comments sentiment analysis

```

from itertools import islice
from youtube_comment_downloader import *
downloader = YoutubeCommentDownloader()

# Open the file
file = open("IDYoutube2.txt", "r", encoding="utf-8")

# Read each line of the file
for line in file:
    # Process each line as needed
    text = "https://www.youtube.com/watch?v=" + line.strip()
    comments = downloader.get_comments_from_url(text , sort_by=SORT_BY_POPULAR)
    #print(line.strip()) # Print the line without extra whitespace
    i=1
    with open('exemple2.txt', 'a',encoding="utf-8") as f:
        for comment in islice(comments, 200000):
            f.write(comment["text"] + '\n')

# Close the file
file.close()

```

Figure 3. The code for collecting data from YouTube comments using Youtube-Comment-Downloader

In addition to BeautifulSoup, we utilized two Python libraries, `itertools` and `Youtube-Comment-Downloader`, to further facilitate the data collection process. `Itertools`, a built-in Python library, provided functions and iterators that aided in efficient iteration and looping, enhancing the overall data retrieval process. `Youtube-Comment-Downloader`, a third-party library specifically designed for downloading comments from YouTube videos, simplified interaction with the YouTube API and enabled the retrieval of comments based on video URLs as demonstrated in Figure 3.

By leveraging these libraries, our code effectively processed each line from a file, constructed YouTube video URLs, retrieved comments using `Youtube-Comment-Downloader`, and appended them to an output file. This automated approach ensured the comprehensive collection of comments from TED talks videos on YouTube, laying the foundation for our sentiment analysis task.

b) Step 2: Pre-Processing

After collecting the dataset of comments from websites and YouTube, it is important to preprocess the data before performing sentiment analysis. The preprocessing stage plays a vital role in text classification by ensuring that the data is clean and optimized for analysis. When working with online texts, it is common to encounter noise and irrelevant elements such as HTML tags, scripts, and advertisements. These elements can negatively impact the accuracy of sentiment analysis. Additionally, not all words in a text carry significant meaning or contribute to its overall sentiment. By identifying and eliminating these uninformative words, the preprocessing step helps refine the data and improve the efficiency of the classification process.

In our research, we utilized the `spaCy` library [17] for data cleaning purposes. We employed stemming to extract word roots, eliminated stop words to reduce word count, converted all words to lowercase, and removed both punctuation and whitespace.

The following are some common preprocessing steps that can be performed on the collected dataset:

- ✓ **Lowercasing:** ensures the model treats identical words regardless of case.

- ✓ **Tokenization:** This involves breaking the text down into individual words, phrases, or sentences, which are then treated as separate units for analysis as detailed in Figure 4.

- ✓ **Stopword removal:** Stopwords are common words such as "and", "the", and "is" that do not add any meaning to the text. The efficiency of the model can be improved by eliminating these words and reducing the dimensionality of the data as illustrated in Figure 5.

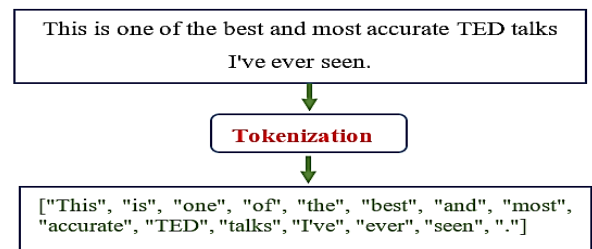


Figure 4. Tokenization

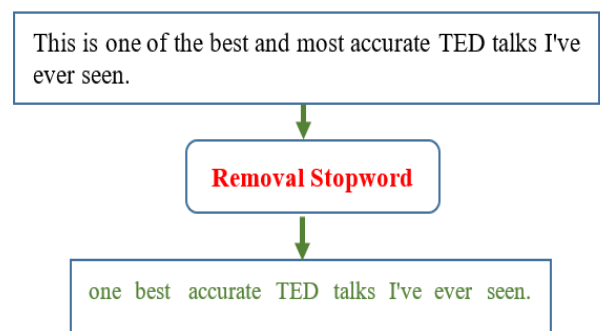


Figure 5. Removal stopword

- ✓ **Stemming/Lemmatization:** Stemming involves reducing words to their root form (e.g., "running" to "run"), while lemmatization involves reducing words to their base form (e.g., "ran" to "run"). This step helps to reduce the number of unique words in the dataset, which can improve the performance of the model.

✓ **Removing special characters and punctuation:** Special characters and punctuation can cause noise in the data and may not add any value to the analysis. Removing them can help to make the data cleaner and easier to analyze.

✓ **Removing duplicates:** Duplicate comments can skew the results of the analysis, so it is important to remove them before analysis.

✓ **Spell checking:** Correcting spelling errors can help to improve the accuracy of the analysis.

c) Step 3: Measurement and Operationalization of Sentiment Variables

In our study, we employed a text analysis approach to measure sentiment in TED talk comments. Specifically, we utilized the TextBlob library in Python to assign sentiment polarity to each comment. TextBlob is a natural language processing library that provides tools for sentiment analysis by evaluating the polarity—whether positive, negative, or neutral—of each text. By using this method, we quantified the sentiment expressed in the comments and operationalized it for our analysis. This approach enabled us to effectively capture variations in sentiment within TED talk comments, thereby providing a robust foundation for our sentiment analysis study. The polarity score provided by TextBlob ranges from -1 to 1, where scores below 0 represent a negative sentiment, scores above 0 represent a positive sentiment, and a score of 0 indicates a neutral sentiment.

$$Table\ T = \begin{cases} Positive, & P_i > 0 \\ Negative, & P_i < 0 \\ Neutral, & P_i = 0 \end{cases} \quad (1)$$

Table 1 also shows the number of comments in each class. We have a total of 117455 comments: 21506 negative, 42914 neutral and 53035 positives as shown in Table 1 and Figure 6.

Table 1. The number of comments in each class

Technique	Positive	Negative	Neutral	Total
TextBlob	53035	21506	42914	117455

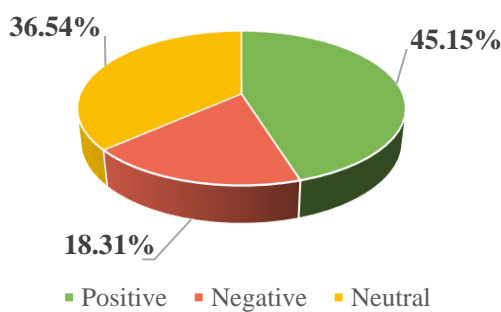


Figure 6. The results of the sentiment analysis

d) Step 4: Feature Extraction

In the proposed system, the feature extraction process involves using CountVectorizer from the sklearn library [18], which produces a sparse matrix that captures all the words contained in each document.

In this scenario, every vector contains all the words that appear in the document, thereby expanding the size of the matrix. Nevertheless, the max-count parameter of CountVectorizer can be utilized to restrict the number of

features.

e) Step 5: Classification Process:

The data is split into two sets: 80% of the data is used to train the algorithm and the remaining 20% is used to test the trained model. A model is created for each ML algorithm and trained with the data. After the learning process, the results are checked with the test data to evaluate the performance of the model. This task is replicated with five distinct ML methods. The models used in this study are discussed in detail in the next section.

f) Step 6: Output:

This marks the final stage in the architecture, whereby sentiment analysis process is concluded by categorizing the TEDX talks comments.

4. CLASSIFICATION MODELS

In the domain of machine learning, classification techniques have been advanced, which employ different approaches to classify labeled and unlabeled data. Some classifiers may require training data to effectively categorize new instances. Noteworthy examples of supervised machine learning classifiers include NB, RF, and SVM [19, 20]. Training a classifier effectively plays a crucial role in simplifying future predictions.

4.1 Naïve bayes

This algorithm is frequently employed in ML techniques and is commonly utilized in text classification and sentiment analysis. Its popularity is due to its speed, simplicity, and effectiveness, making it a preferred method for sentiment analysis in a various sectors including marketing, politics, and social media.

4.2 Random Forest

Random Forest [21] is a method designed to improve prediction accuracy and model robustness by aggregating numerous decision trees. It achieves this by training each decision tree on random subsets of the training data and aggregating their predictions. Random Forest incorporates randomness in two keyways. Firstly, at each node of the decision tree, it randomly selects subsets of features for splitting, effectively reducing overfitting and introducing diversity among the trees. Secondly, it employs bootstrap sampling, where each tree is trained on random subsets of the data with replacement. This introduces additional diversity and helps reduce variance in the model. Random Forest is well-suited for handling large datasets, noisy data, and offers insights into the importance of different features. It finds extensive applications in machine learning for classification and regression tasks.

4.3 K-nearest neighbor (KNN)

KNN, a ML algorithm utilized for supervised learning in both classification and regression tasks, is recognized for its non-parametric nature, relying on data point similarities for predictions. The parameter "K" in KNN represents the number of nearest neighbors considered for prediction. The algorithm

calculates the distance between a test data point and all training data points, then selects the K nearest neighbors. In classification, the majority class among these neighbors determines the predicted class for the test data point, while in regression, the predicted value is the average of the target values of the K nearest neighbors. Renowned for its simplicity and intuitive approach, KNN finds applications across diverse domains including text classification, image recognition, and recommender systems.

4.4 Support vector machine (SVM)

The main objective of SVM is to identify the best hyperplane for separating data points into separate classes, usually a positive class and a negative class. In binary classification scenarios, SVM tries to find a hyperplane that maximizes the margin between these classes [refer to Figure 7]. The margin represents the distance between the hyperplane and the nearest data points from each class. Furthermore, SVMs can handle non-linearly separable data by utilizing kernel functions to make it linearly separable in a higher-dimensional feature space.

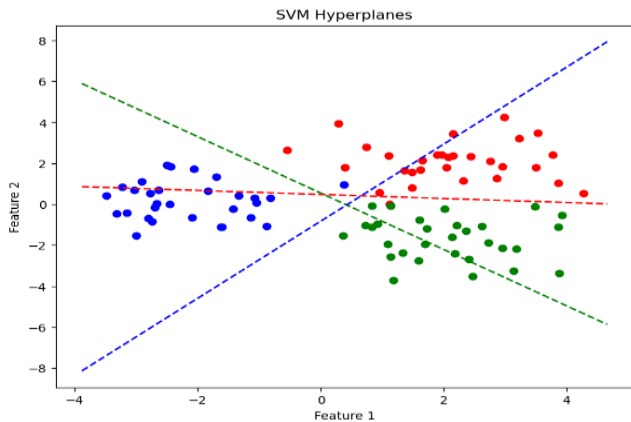


Figure 7. SVM hyperplanes

4.5 Decision tree

This classifier begins at the root and splits the data based on the attribute that results in the highest information gain (IG). This process is repeated until all the leaf nodes are pure, meaning they contain instances of only one class. To prevent overfitting, a depth limit can be set for the tree, ensuring a more generalized model. To create a decision tree model, training data is used, and various validation techniques are employed to assess and enhance its performance like [22, 23]. There are primarily two types of decision trees: classification trees, where the decision variable is discrete and clearly defined, and regression trees, where the decision factors take continuous values.

In summary, the integration of theoretical foundations with machine learning models in SA offers a comprehensive approach to understanding and interpreting sentiments expressed in textual data. Theoretical frameworks from affective computing, linguistic analysis, and cognitive psychology provide insights into the cognitive and emotional aspects of language, laying the groundwork for sentiment analysis methodologies. By aligning with these theoretical principles, machine learning models like NB, RF, K-NN, SVM, and DT effectively capture and interpret sentiment patterns in text. From probabilistic reasoning to ensemble learning and

similarity-based approaches, each model leverages theoretical concepts to enhance sentiment classification accuracy and robustness. Through concrete examples and applications across various domains, these models demonstrate their alignment with theoretical principles and their efficacy in real-world sentiment analysis tasks, bridging the gap between theory and practice in sentiment analysis.

5. EXPERIMENTAL RESULT

This study encompasses the application of five different ML techniques to determine the sentiment polarity of TED talks comments. The performance of the various classifiers has been assessed using metrics:

$$Precision = tp / (tp + fp)$$

$$Recall = tp / (tp + fn)$$

$$F1\text{-score} = 2 \cdot 1 / (1 / recall + 1 / precision) = 2 \cdot (precision \cdot recall / (precision + recall))$$

True positives (*tp*), false positives (*fp*), and false negatives (*fn*) used to describe the number of positive instances that were correctly identified, incorrectly identified positive instances, and incorrectly identified negative instances. The accuracy of the classifier in identifying positive instances is measured by the proportion of true positives among all positive estimates, which is known as precision. A higher precision implies a lower rate of false positives, while a lower precision indicates a higher rate of false positives. Alternatively, recall evaluates the classifier's sensitivity by determining the percentage of positive instances correctly recognized. A higher recall indicates a reduced occurrence of false negatives. In simpler terms, recall indicates the fraction of accurately classified instances from the entire predicted instances.

The F1-score measure, resulting from the combination of precision and recall, serves as a unified metric that evaluates a classifier's performance. It represents the balanced harmonic mean of precision and recall, taking into account the classifier's ability to accurately identify positive instances while considering the coverage of all relevant instances.

In this comparative analysis of different classifiers, we evaluated their efficacy employing diverse metrics including accuracy, precision, recall, and F1-score. The outcomes demonstrate that SVM showcased the highest performance, achieving an accuracy of 94.93%, precision of 94.92%, recall of 94.93%, and an F1-score of 94.84% among all classifiers. The values for all metrics can be found in Table 2 and Figure 8.

Table 2. The values of the metrics for each classifier

Classifier	Accuracy	Precision	Recall	F1-Score
KNN	46.01%	71.02%	46.01%	38.90%
SVM	94.93%	94.92%	94.93%	94.84%
Random Forest	88.82%	89.63%	88.82%	88.10%
Naïve Bayes	67.50%	77.95%	67.50%	62.86%
Decision tree	89.63%	89.48%	89.63%	89.53%

Random Forest also performed well with an accuracy of 88.82%, precision of 89.63%, recall of 88.82%, and F1-score of 88.10%. Decision Tree achieved an accuracy of 89.63%,

precision of 89.48%, recall of 89.63%, and F1-score of 89.53%. KNN (K-Nearest Neighbors) showed relatively lower performance with an accuracy of 46.01%, precision of 71.02%, recall of 46.01%, and F1-score of 38.90%. Naïve Bayes achieved an accuracy of 67.50%, precision of 77.95%, recall of 67.50%, and F1-score of 62.86%. These findings suggest that SVM and Random Forest are promising classifiers for the given task, while KNN and Naïve Bayes exhibit lower performance. The decision tree classifier also performs well and provides a balanced performance across all metrics. It's important to consider the specific dataset and problem domain when selecting the appropriate classifier for sentiment analysis.

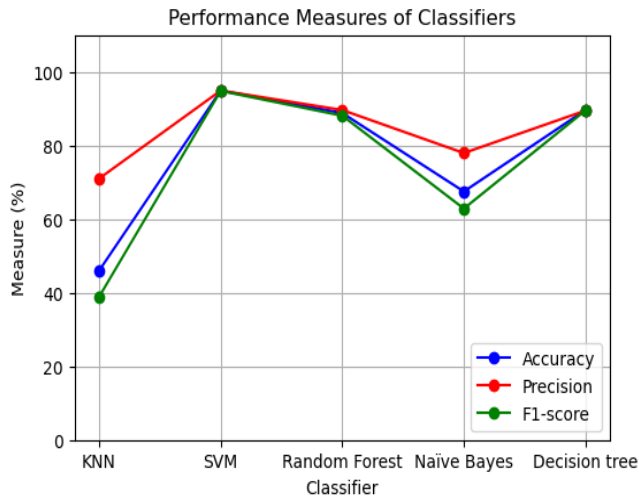


Figure 8. The performance measures of classifiers

6. CONCLUSIONS

The primary objective of this investigation was to apply diverse machine learning techniques for sentiment polarity extraction from TED talks comments. The research involved implementing and evaluating the effectiveness of five classifiers, namely Naive Bayes, SVM, Random Forest, KNN, and Decision Tree. The outcomes of the study suggested that SVM delivered the most precise results. It is important to note that only the fundamental forms of these classifiers were used, indicating that further accuracy improvements are possible by adjusting their parameters.

The findings of our research could have significant implications for areas such as digital marketing, educational technology, and data analytics. In digital marketing, understanding sentiment patterns can inform targeted advertising strategies and enhance customer engagement efforts. Similarly, sentiment analysis can play a crucial role in personalized learning experiences within educational technology by gauging student sentiments and adapting instructional content accordingly. Furthermore, in data analytics, sentiment analysis can complement traditional data analysis methods, providing valuable insights into consumer behavior and market trends. Acknowledging limitations in data collection, analysis, and interpretation is vital. These constraints may have affected the reliability and validity of the findings. To address them, solutions like expanding data sources and using advanced analytical techniques can be applied, enhancing the study's methodological robustness and interdisciplinary relevance in sentiment analysis. By bridging the gap between sentiment analysis and these interdisciplinary

areas, we can maximize the utility of our research and contribute to the advancement of multiple domains. Our study has also advanced knowledge in sentiment analysis within the context of YouTube comments and TED talks. While focusing on these platforms, we've set the stage for future research to explore sentiment across diverse domains, leveraging aspect-based analysis and ontology-driven approaches. By acknowledging limitations and suggesting future directions, we aim to fuel ongoing innovation in sentiment analysis, enriching our understanding of nuanced sentiment expressions across varied contexts.

REFERENCES

- [1] Mao, Y., Zhang, Y., Jiao, L., Zhang, H. (2022). Document-level sentiment analysis using attention-based Bi-directional Long Short-Term Memory Network and two-dimensional convolutional neural network. *Electronics*, 11(12): 1906. <https://doi.org/10.3390/electronics11121906>
- [2] Liu, F., Zheng, L., Zheng, J. (2020). HieNN-DWE: A hierarchical neural network with dynamic word embeddings for document-level sentiment classification. *Neurocomputing*, 403: 21-32. <https://doi.org/10.1016/j.neucom.2020.04.084>
- [3] Choi, G., Oh, S., Kim, H. (2020). Improving document-level sentiment classification using importance of sentences. <https://doi.org/10.48550/arXiv.2103.05167>
- [4] Al Amrani, Y., Lazaar, M., El Kadiri, K.E. (2018). Random forest and support vector machine based hybrid approach to sentiment analysis. *Procedia Computer Science*, 127: 511-520. <https://doi.org/10.1016/j.procs.2018.01.150>
- [5] Srujan, K.S., Nikhil, S.S., Raghav Rao, H., Karthik, K., Harish, B., Keerthi Kumar, H. (2018). Classification of amazon book reviews based on sentiment analysis. In V. Bhateja, B. Nguyen, N. Nguyen, S. Satapathy, & D. N. Le (Eds.), *Information Systems Design and Intelligent Applications. Advances in Intelligent Systems and Computing*, Vol. 672, pp. 221-229. https://doi.org/10.1007/978-981-10-7512-4_40
- [6] Bayhaqy, A., Nainggolan, K., Sfenrianto, S., Kaburuan, E. R. (2018). Sentiment analysis about e-commerce from tweets using decision tree, k-nearest neighbor, and naïve bayes. In *Proceedings of the International Conference on Technology, Informatics, Management, Engineering, and Environment, Nusa Dua, Bali, Indonesia*, pp. 1-6. <https://doi.org/10.1109/ICOT.2018.8705796>
- [7] Tripathy, A., Agrawal, A., Rath, S.K. (2015). Classification of sentimental reviews using machine learning techniques. *Procedia Computer Science*, 57: 821-829. <https://doi.org/10.1016/j.procs.2015.07.523>
- [8] Amini Motlagh, M., Shahhoseini, H., Fatehi, N. (2023). A reliable sentiment analysis for classification of tweets in social networks. *Social Network Analysis and Mining*, 13(7). <https://doi.org/10.1007/s13278-022-00998-2>
- [9] Erfina, A., Rifki Nurul, M. (2023). Implementation of naïve bayes classification algorithm for twitter user sentiment analysis on chatgpt using python programming language. *Data & Metadata*, 2: 45. <https://doi.org/10.56294/dm202345>
- [10] Gupta, K., Jiwani, N., Afreen, N. (2023). A combined approach of sentimental analysis using machine learning

- techniques. *Revue d'Intelligence Artificielle*, 37(1): 1-6. <https://doi.org/10.18280/ria.370101>
- [11] Saha, S., Yadav, J., Ranjan, P. (2017). Proposed approach for sarcasm detection in twitter. *Indian Journal of Science and Technology*, 10(25): 1-8.
- [12] Borg, A., Boldt, M. (2020). Using VADER sentiment and SVM for predicting customer response sentiment. *Expert Systems with Applications*, 162: 113746. <https://doi.org/10.1016/j.eswa.2020.113746>
- [13] Ain, Q. T., Ali, M., Riaz, A., Noureen, A., Kamran, M., Hayat, B., Rehman, A. (2017). Sentiment analysis using deep learning techniques: A review. *International Journal of Advanced Computer Science and Applications*, 8(6): 424-433. <https://doi.org/10.14569/IJACSA.2017.080657>
- [14] Habimana, O., Li, Y., Li, R., Gu, X., Yu, G. (2020). Sentiment analysis using deep learning approaches: An overview. *Science China Information Sciences*, 63(1): 1-36. <https://doi.org/10.1007/s11432-018-9941-6>
- [15] Ba Alawi, A., Bozkurt, F. (2024). A hybrid machine learning model for sentiment analysis and satisfaction assessment with Turkish universities using Twitter data. *Decision Analytics Journal*, 11: 100473. <https://doi.org/10.1016/j.dajour.2024.100473>
- [16] Lasri, I., Riadsolh, A., ElBelkacemi, M. (2023). Self-attention-based bi-LSTM model for sentiment analysis on tweets about distance learning in higher education. *International Journal of Emerging Technologies in Learning*, 18(12): 119-141. <https://doi.org/10.3991/ijet.v18i12.38071>
- [17] Bansal, B., Srivastava, S. (2018). Sentiment classification of online consumer reviews using word vector representations. *Procedia Computer Science*, 132: 1147-1153. <https://doi.org/10.1016/j.procs.2018.05.029>
- [18] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., Louppe, G. (2012). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825-2830.
- [19] Sabouri, Z., Gherabi, N., Nasri, M., Amnai, M., El Massari, H., Moustati, I. (2023). Prediction of depression via supervised learning models: Performance comparison and analysis. *International Journal of Online and Biomedical Engineering*, 19(9): 93-107. <https://doi.org/10.3991/ijoe.v19i09.39823>
- [20] Qanouni, F., Ghandi, H., Gherabi, N., Massari, H.E. (2024). Machine learning models for detection COVID-19. In N. Gherabi, A. I. Awad, A. Nayyar, & M. Bahaj (Eds.), *Advances in intelligent system and smart technologies*, pp. 95-108. Springer. https://doi.org/10.1007/978-3-031-47672-3_12
- [21] Breiman, L. (2001). Random forests. *Machine Learning*, 45: 5-32. <https://doi.org/10.1023/A:1010933404324>
- [22] Massari, H.E., Gherabi, N., Mhammedi, S., Ghandi, H., Qanouni, F., Bahaj, M. (2022). Integration of ontology with machine learning to predict the presence of covid-19 based on symptoms. *Bulletin of Electrical Engineering and Informatics*, 11(5): 2805-2816. <https://doi.org/10.11591/eei.v11i5.4392>
- [23] Massari, H.E., Gherabi, N., Mhammedi, S., Sabouri, Z., Ghandi, H. (2022). Ontology-based decision tree model for prediction of cardiovascular disease. *Indian Journal of Computer Science and Engineering*, 13(3): 851-859. <https://doi.org/10.21817/indjcse/2022/v13i3/221303143>